Final Technical Report for ONR Grant N00014-89-J-1064

1 October 1988 to 31 January 1994

George S. Avrunin Jack C. Wileden

1 Summary of Technical Accomplishments

This project investigated the problem of analyzing concurrent and distributed systems, in order to determine whether they behave as intended by their developers. We explored analysis of both "logical" properties, such as freedom from deadlock or enforcement of mutually exclusive access to a resource, and timing properties, such as the time that can elapse between the occurrence of certain events in an execution of the system. Our work has focussed on the development of automated analysis techniques that could serve as the basis for practical tools to be used by developers of concurrent systems.

The major difficulty in analyzing the behavior of concurrent systems is the combinatorial explosion in the number of possible states of the systems as the number of component processes increases. The approach taken in this project deals with the state space explosion by attempting to find strong necessary conditions, in the form of linear inequalities, for there to exist an execution of the concurrent system with a certain property and using standard integer programming techniques to determine whether these necessary conditions are consistent [3,8]. (References in this section refer to the publications listed in the next section. Additional references to earlier work, and the work of other investigators, can be found in those papers.)

At the start of this project, we had designed a prototype toolset based on these techniques. One of the first tasks of this project was an initial implementation of that toolset, together with experimental application of it to a variety of small concurrent and real-time systems [9, 10]. These experiments indicated that the method could be used with systems that were large by then-current standards for automated analysis. We then reimplemented the toolset to incorporate insights gained from experience with the original version and new analysis techniques [2]. Experiments with this new version [4–6] showed that the techniques could be successfully applied to concurrent systems with hundreds of processes.

Later work in the project focussed on extending the range of problems to which our methods could be successfully applied. In the course of this work, components of the toolset have been re-implemented to improve efficiency and implement new analysis techniques or add other functionality. In particular, our former student, James C. Corbett, who is now at the University of Hawaii, reimplemented the component of our toolset that translates concurrent system specifications to finite state automata. His new "deriver" supports a number of new features in the specification lagnuage, allowing the convenient description of a wider range of systems, and uses data-flow analysis techniques to prune the automata. It therefore produces smaller automata, allowing larger systems to be analyzed. Using this new tool, we have carried out timing analyses of uniprocessor concurrent systems with more than 2^{500} reachable states [7].

Corbett and Avrunin have also developed and implemented methods for finding bounds on the time between events in multiprocessor concurrent systems [14]. They are currently working on improving the efficiency of these methods, which have successfully been applied to multiprocessor systems with more than 100 concurrent processes. Preliminary results suggest that new implementations of these methods will be useful with systems at least 2 or 3 times larger, having more than 2^{300} reachable states. Corbett and Avrunin [16] have also described an extended version of the basic constrained expression analysis method (based in part on Corbett's Ph.D. dissertation) and given an analysis of its expressive power.

In other work, Avrunin has been collaborating with Professor Victor Yodaiken on the development of compositional methods that would allow large systems to be handled by analyzing subsystems separately, using constrained expression techniques, model checking, or some other method, and then combining the results of these analyses [17]. This work uses Yodaiken's modal primitive recursive function approach to provide both a very general composition operator and concise descriptions of very large state machines. Corbett and Avrunin [15] have investigated the direct application of the inequality-based techniques developed in this project to show that a component of a large system is equivalent, in the sense of having the same external behavior, to a simpler subsystem, allowing the use of the simpler version in analysis.

One of our interests in this project was to begin to develop a framework for characterizing classes of concurrent and real-time systems problems and analysis techniques. Such a framework would provide a basis for understanding, organizing and comparing the capabilities and results obtained by various existing or proposed approaches to analysis of concurrent and real-time systems. This will be important both as an abstract, scientific contribution to taxonomizing an area of computer science research and also as a practical aid to developers of concurrent and real-time software who may be faced with choosing among alternative analysis methods for application to a given analysis problem. Wileden has been working on the definition of a formal basis for such a characterization framework. One aspect of this effort has involved extensive experimentation with application of our constrained expression analysis toolset to a few standard concurrency analysis problems, such as several variations on the dining philosophers problem. This experimentation has yielded some insights, and also some intriguing puzzles, regarding the dimensions of variability in how susceptible seemingly very similar problems are to analysis using our automated methods. Another aspect of this work has been continued exploration of the use of our constrained expressions formalism as a formal basis for describing features of systems and analysis problems. This approach continues to show promise, but additional investigation, based on data from analysis of further examples using both our constrained expression tools and other techniques, will be needed to validate his initial observations and to extend and refine the framework.

2 Publications Supported by this Grant

[1] G. S. Avrunin. Sharpening bounds on the time between events in maximally parallel systems. Technical Report 92-69, Department of Computer Science, University of Massachusetts at Amherst, 1992. Available for anonymous ftp on ext.math.umass.edu.

[2] G. S. Avrunin, U. Buy, and J. C. Corbett. Automatic generation of inequality systems for constrained expression analysis. Technical Report 90-32, Department of Computer and Information Science, University of Massachusetts, Amherst, 1990.

[3] G. S. Avrunin, U. A. Buy, and J. C. Corbett. Integer programming in the analysis of concurrent systems. In K. G. Larsen and A. Skou, editors, *Computer Aided Verification, 3rd International Workshop Proceedings*, volume 575 of *Lecture Notes in Computer Science*, pages 92–102, Aalborg, Denmark, July 1991. Springer-Verlag.

[4] G. S. Avrunin, U. A. Buy, J. C. Corbett, L. K. Dillon, and J. C. Wileden. Automated analysis of concurrent systems with the constrained expression toolset. *IEEE Trans. Softw. Eng.*, 17(11):1204–1222, Nov. 1991.

[5] G. S. Avrunin, U. A. Buy, J. C. Corbett, L. K. Dillon, and J. C. Wileden. Experiments with an improved constrained expression toolset. In *Proceedings of the Symposium on Testing, Analysis, and Verification (TAV4)*, pages 178–187, New York, October 1991. Association for Computing Machinery.

[6] G. S. Avrunin, J. C. Corbett, L. K. Dillon, and J. C. Wileden. Automated constrained expression analysis of real-time software. Technical Report 90-117, Department of Computer Science, University of Massachusetts at Amherst, 1992.

[7] G. S. Avrunin, J. C. Corbett, L. K. Dillon, and J. C. Wileden. Automated derivation of time bounds in uniprocessor concurrent systems. *IEEE Trans. Softw. Eng.*, 20(9):708–719, Sept. 1994.

[8] G. S. Avrunin, L. K. Dillon, and J. C. Wileden. Constrained expression analysis of real-time systems. Technical Report 89-50, Department of Computer and Information Science, University of Massachusetts, 1989.

[9] G. S. Avrunin, L. K. Dillon, and J. C. Wileden. Experiments with automated constrained expression analysis of concurrent software systems. In R. A. Kemmerer, editor, *Proceedings of the ACM SIGSOFT '89 Third Symposium on Software Testing, Analysis and Verification*, pages 124–130, December 1989. Appeared as *Software Engineering Notes*, 14(8).

[10] G. S. Avrunin and J. C. Wileden. Improvements in automated analysis of concurrent and real-time software. In A. M. van Tilborg and G. M. Koob, editors, *Foundations of Real-Time Computing: Formal Specifications and Methods*, chapter 8, pages 195–215. Kluwer Academic Publishers, 1991.

[11] J. C. Corbett. Automated formal analysis methods for concurrent and real-time software. Technical Report 92-48, Department of Computer Science, University of Massachusetts at Amherst, 1992.

[12] J. C. Corbett. Verifying general safety and liveness properties with integer programming. In G. v. Bochmann and D. K. Probst, editors, *Computer Aided Verification, 4th International Workshop*, volume 663 of

Lecture Notes in Computer Science, pages 357–369, Montreal, Canada, 1992. Springer-Verlag.

[13] J. C. Corbett. Identical tasks and counter variables in an integer programming based approach to verification. In M. Feather and A. van Lamsweerde, editors, *Proceedings of the Seventh International Workshop on Software Specification and Design*, pages 100–109, Los Alamitos, California, Dec. 1993. IEEE Computer Society Press.

[14] J. C. Corbett and G. S. Avrunin. A practical method for bounding the time between events in concurrent real-time systems. In T. Ostrand and E. Weyuker, editors, *Proceedings of the 1993 International Symposium on Software Testing and Analysis (ISSTA)*, pages 110–116, Cambridge, MA, June 1993. ACM Press (Proceedings appeared in *Software Engineering Notes*, 18(3)). An updated version is available for anonymous ftp on ext.math.umass.edu.

[15] J. C. Corbett and G. S. Avrunin. Towards scalable compositional analysis. In D. Wile, editor, *Proceedings of the Second ACM SIGSOFT Symposium on Foundations of Software Engineering*, pages 53–61, New Orleans, Dec. 1994. ACM Press (Proceedings appeared in *Software Engineering Notes*, 19(5)).

[16] J. C. Corbett and G. S. Avrunin. Using integer programming to verify general safety and liveness properties. *Formal Methods in System Design*, to appear.

[17] V. Yodaiken and G. S. Avrunin. Real-time state machines and circuit verification with modal functions. Technical Report 93-04, Department of Computer Science, University of Massachusetts at Amherst, 1993.

3 Software Prototypes Developed

Our techniques have been implemented in a series of prototype toolsets. The earliest version consisted of five separate components, written in Ada, FORTRAN, and Lisp. In more recent versions, four of these components have been integrated into a single Lisp program that generates inequalities starting from a description of the concurrent system given in an Ada-like design language and then interprets solutions to those inequalities. The systems of inequalities are solved by an integer programming package, written in FORTRAN and based on the MINOS optimization system from Stanford. The new Lisp program, developed by Corbett in the course of his dissertation work (which was supported by this project) and more recent work at the University of Hawaii, supports additional features in the specification language and permits much more compact specifications. It also incorporates data-flow analysis techniques to prune the automata, thereby allowing larger systems to be analyzed. Avrunin and Corbett have also modified this tool to implement several new analysis techniques for real-time systems, as well as their compositional techniques.

4 Transitions

Professor James C. Corbett, of the University of Hawaii at Manoa, is actively involved in research on real-time systems with Professor Avrunin, and has recently extended some of their analysis methods and tools to handle the full real-time capabilities of Ada 9X. Our results and prototype software tools have been shared with Professor Laura Dillon at the University of California, Santa Barbara, whose work involves both constrained expression analysis and the use of interval logic in the analysis of real-time systems. Dillon is currently using some of the new constrained expression-based methods developed by Corbett and Avrunin to verify assertions in real-time interval logic. At the request of Professor Sol Shatz of the University of Illinois at Chicago, Avrunin and Corbett have analyzed a number of example systems in order to provide comparisons between the constrained expression methods and Shatz's reachability space reduction methods.